Overcoming the obstacles of harvesting and searching digital repositories from federated searching toolkits, and embedding them in VLEs

Santiago Chumbe[1], Roddy MacLeod[2], Phil Barker[1], Malcolm Moffat[1], and Roger Rist[1]

[1]Institute for Computer Based Learning, School of Mathematical and Computer Sciences, HeriotWatt University, Edinburgh, EH14 4AS, UK

[2]Heriot-Watt University Library
HeriotWatt University, Edinburgh, EH14 4AS, UK

Abstract:
This paper addresses two important needs. The first one is the need to alleviate the resource discovery task across digital repositories by subject, which includes the ability of searching heterogeneous sources that apply to a specific audience (e.g. engineering academics) or purpose (e.g. research, teaching) from one access point. The second need is to provide toolkits for federated searching which are able to be embedded in electronic learning environments used by lecturers, students and researchers. Most of these environments are institutional Virtual Learning Environments (VLEs) and Portals. Our study will show that the satisfaction of both needs faces important obstacles. On one side, standard exchange formats such as Z39.50 or OAI, developed precisely to facilitate the transfer or sharing of data between computer systems, present obstacles that make the harvesting and searching of data from digital repositories a challenging process. On the other side, VLEs are often restricted in their ability to allow the sharing and re-use of external e-learning sources discovered by federated searching toolkits. A solution for these obstacles, based on a service-oriented architecture approach, is suggested and explored on a pilot system. The aim of our research is the realisation of the concept of flexible federated searching. The intention is that the VLE user should be able to use whatever search tool he/she likes for whatever repositories he/she needs to search, without concern for how the tool and the repositories manage to communicate, or how the tool makes search results available to other VLE components. The pilot system attempts to demonstrate that most of the flexible federated searching concept can be achieved by making proper use of current interoperability standards for digital repositories and e-learning systems.

Keywords:
Digital Repositories, Federated Searching, Z39.50, OAI-PMH, VLE, Web Services.

1. Introduction

Federated Searching has long been a desirable goal for the scholarly community. Most early implementations of federated search services used the Z39.50 protocol to perform distributed searches across remote databases. More recently, the creation of real federated-type searching services has benefited from an increasing number of OAI metadata providers, and the availability of Web-based specifications such as the

SRU/SRW protocols. In the following paragraphs we give a brief presentation of the Z39.50, OAI and SRU/SRW protocols, and put our work in context by introducing the specific information environment we have to deal with.

Z39.50 is a pre-Web protocol for searching and retrieving information from remote databases [1]. Z39.50-based services are widely incorporated into library systems. A typical Z-39.50 session goes through *Initialisation* (connection, negotiating levels of service), *Search* (sending query and getting back result set) and *Retrieval* (retrieval of records from the result set as specified by client.)

The Open Archives Initiative (OAI) protocol is a low-barrier interoperability solution to access across fairly heterogeneous repositories. The OAI-Protocol for Metadata Harvesting (OAI-PMH) defines a mechanism for harvesting records containing metadata from repositories via HTTP [2].

SRU (Search/Retrieve via URL) is a protocol where the search queries are sent to the SRU server encapsulated in HTTP URLs. SRW (Search Retrieve Web Service) is a companion protocol to SRU. With SRW the messages are conveyed from client to server, not by a URL, but instead using XML over HTTP via SOAP [3].

We are using these protocols in an academic context, but the information environment we have to deal with is quite diverse, ranging from institutional repositories to commercial proprietary databases. The technology of our work basically consists of cross-searching distributed Z39.50 repositories together with metadata previously harvested from OAI digital repositories. As the importance of subject access to information has been recognised in the literature [4], our work also intends to pilot a distributed subject model, with specific reference to engineering. In this way, it is hoped that the resulting pilot service will potentially satisfy some information retrieval needs of the engineering community and take account of the advice of authors such as Stephen and Harrison (2002): *"Electronic services need to be designed differentially and should deploy technologies selectively in service of the varying scholarly practices that define different fields."*

In the next sections we will discuss the issues associated with the Z39.50, OAI and SRU/SRW protocols and their implementations. Also, we will discuss the obstacles we have found for embedding federated search toolkits within two popular VLEs. Finally we will present the PerX pilot system for federated searching which is being developed at Heriot Watt University as a part of a research project of the JISC-funded Digital Repositories Programme. We will discuss alternatives to overcome the identified obstacles, making use of the experiments done with the pilot system. The pilot includes a federated search toolkit via XML, which makes it potentially embeddable within institutional VLEs that have a service-oriented architecture. We will use the outcomes of this pilot to draft our conclusions.

2. Identified issues associated with the Z39.50, OAI and SRU/SRW protocols

Efficient and effective implementation of federated searching is a complex task due to the tremendous growth in the number of digital resource repositories and, ironically, due to the concurrent development of many different metadata standards. In this section, we will briefly describe the issues and technical obstacles presented by each of these protocols.

2.1. Issues of Z39.50 Distributed Searching

Z39.50 architecture limits the possible ways of accessing and processing data, and thereby more or less predefines the service functions we can achieve. In a Z39.50 Distributed Searching architecture, queries expressed in PQF[1] are passed directly from the query engine to the remote Z39.50 server that abstracts from the specific implementation of the repository. Consequently, federated searching based on Z39.50 does not provide full control of the result sets returned by these remote repositories.

Next follows an overview of obstacles that are inherent to the Z39.50 protocol, which we must deal with when searching Z39.50 repositories. Our overall impression is that Z39.50 is a protocol that cannot completely meet the expectations of a federated service that aims to alleviate the resource discovery task across digital repositories. Only a full knowledge of the Z39.50 repositories to be searched, and appropriate tuning between the configuration settings of the repositories and the federated search software, can deliver a limited, but still useful, service for discovering data held in heterogeneous sources.

The first issue faced by a federated service that makes use of the Z39.50 protocol is that Z39.50 repositories can be implemented in different ways. In order to abstract from this technical heterogeneity, our software middleware needs to be able to generate different PQF syntaxes to abstract the queries into vectors of 'attributes.' Generating these vectors is not a trivial task. One uncertain but still common and simple solution is simply to rely on the default settings of the Z39.50 server provider and hope that they match our requirements. But, how does one establish what the default settings are? The Z39.50 offers the *Explain* facility, but in practice this feature is not of sufficient help, as most of the Z39.50 clients do not support this facility and even when available, it is frequently unable to give full information on how the server has implemented the Z39.50 protocol.

The second challenge of distributed architecture is that information relevant to a query is distributed over the different sources. Ideally, after locating and retrieving relevant information from multiple repositories, we would like to be able to remove duplications and combine the individual results. The Z39.50 architecture does not facilitate the implementation of that functionality. The aim of the Z39.50 standard has been to create a single interface from which various Z39.50 databases could be searched. In our opinion, this was probably a very optimistic objective. For example, two UK librarians involved in a project to develop a virtual union catalogue of music libraries came to the realisation that "*each library database has to be configured on an individual basis*" (Hogg and Field, 2001). The reality is that for Z39.50 based federated search services, true de-duplication is virtually impossible and neither can these services perform a relevancy ranking that is globally relevant and uniform for all the searched Z39.50 databases.

A third barrier that we must face is not inherent to the protocol itself, but instead is caused by the Z39.50 repository owners. We have found that most of the available

---

[1] PQF, or Prefix Query Format, is a cryptic notation to generate Z_RPNQuery structures from human-created query notations [5].

Z39.50 servers of interest to our subject were actually incomplete implementations of the Z39.50 protocol. The Z39.50 search specification is very rich, but, although we do not expect that target providers will be prepared to implement all of these options, we have the impression that some providers have opted for “quick and dirty” implementations of the standard, sometimes using incorrect search attributes and incorrect or inadequate diagnostic messages. The issues surrounding misleading implementations of Z39.50 servers have been extensively studied and documented. A widely accepted solution is the development of “profiles”, which are lists of minimum search options and server functionality required by different communities of Z39.50 users. Probably the best known of these is the Bath Profile (Lunau, 2003). Unfortunately this Profile is principally concerned with library OPACs and catalogues, and it does not really address issues relating to other types of databases of interest for federated searching. Furthermore, very few Z39.50 repositories provide support for the Bath Profile. No practical way of enforcing its use has been possible even among the data providers, such as the UK-based Resource Discovery Network (RDN) services [6], which were indirectly involved in the drafting of the Profile.

Database indexing problems also arise out of the use of Z39.50. As Lynch (Lynch, 1997) has pointed out, the Z39.50 protocol is not a database indexing standard, and current Z39.50 attribute sets are not defined in terms of database indexing. As a consequence, Z39.50 vendors have implemented different kind of indexes. The lack of uniformity in database indexing has a negative impact on a federated search services, because their users will receive imprecise search results when the service searches an inappropriate index such as a generic-name index for a specific-author search requests.

Finally we must mention the lack of suitable documentation, because although this is not a critical issue it is still a problem that makes the implementation of federated search services even harder. Z39.50 repositories often provide inadequate, out of date or incomplete documentation, and inaccurate information regarding the configuration of their Z39.50 servers. There is a need for an agreed minimum standard governing the type of really useful information required by services that wish to connect to Z39.50 servers and effectively search them. This has already been noticed in the UK by the Joint Information Systems Committee (JISC), which has funded initiatives such as Information Environment Service Registry (IESR) [7]. One of the aims of IESR is to make it easier for other applications to discover and use Z39.50 targets.

2.2. Issues of OAI Metadata harvesting and transformation

OAI Metadata Repositories are important for their large quantity and rich content. Including an OAI repository in a federated search service involves two main processes, harvesting metadata from the OAI data provider, and transforming suitable data from the harvested metadata. OAI-PMH metadata harvesting relies on machine capabilities to collect XML tagged metadata from remote data providers, via the HTTP protocol. Metadata transformation uses XSLT-based techniques to extract and normalize data contained in the harvested XML required by the service for indexing and searching purposes

The technological challenge with OAI comes when the federated search service needs to deal with metadata providers that do not follow the OAI-PMH standards and recommendations in full. In the next paragraphs we discuss the most common issues that arise when a federated search service use data harvested from OAI repositories.

In an ideal situation, harvesting and extracting data from OAI repositories should be done automatically, without human intervention. The reality is different. Harvesting itself is a process that needs to be monitored from beginning to end. In addition, once we have completed the harvesting, the fact is that we cannot yet trust what we have obtained. We will need to verify the metadata received and we will need software tools to normalize and filter the data relevant for our service.

It is true that once we have harvested an OAI repository for the first time, there is a chance that the next harvests will run with fewer or no problems. However, in practice this is something that rarely occurs, in particular with OAI repositories that have hundreds of thousands of records.

The following are examples of cases where the harvested data needs to be studied, corrected or enhanced before it is included in the service to ensure a certain threshold of usefulness for users.

It is quite common to have to deal with non-valid or/and ill-formed XML documents served by OAI repositories. Such documents cannot be processed automatically by XML-oriented techniques like XSLT unless they are first "cleaned" or debugged by programmatically or manually correcting the errors they contain. This debugging is time consuming and resource intensive, especially since the errors may be specific to particular data sources. An alternative is to abandon the XML-oriented techniques and resort to pattern matching to extract the required data from elements that can be recognised. This approach is also time consuming, especially since one has to ensure that valid variation in the XML (for example the inclusion of white space) does not cause the pattern matching to fail.

It is expected that each record of the OAI repository should include a URL as a unique identifier for the resource. This URL is important because the federated search service will use it to direct the user to the location where the resource is hosted. However, it is not unusual to find OAI records without a URL identifier or with invalid, "aged" or erroneous URLs. If we include them in a service, they will produce the 404 HTTP response code (“Web page not found”). The harvester therefore needs to include a suitable link-checker facility, which is able to detect ill-formed or “aged” URLs included in OAI repositories as unique identifiers.

We have noticed that in many repositories, it is practically impossible to know for sure what kind of digital object is being pointed out from the unique URL provided in the records. This situation is quite common with large OAI repositories, such as *CiteSeer*, that in turn harvest other metadata or data sources. Thus a diversity of digital objects are pointed out from the harvested records of *CiteSeer*, making it difficult to tell users if the provided URL is pointing to a metadata record, a "bridge web-page", the full-text document, or an authentication page, etc. The only effective solution is to contact and work closely with the creators of the OAI repositories, to encourage, for example, the use of richer metadata to include elements beyond the mandatory Dublin Core (DC) schema [8], such as a resource-type element. However, this approach may be unrealistic if the federated search service is harvesting hundreds of OAI repositories.

OAI repositories tend to be black boxes when they are harvested for the first time. For example, we would like to know in advance the number of records that we expect to harvest. This number is especially useful when a large repository is harvested for the first time or there is a need for setting up an automatic harvesting mechanism for periodic updates. Information about the features of the OAI implementation of a repository would allow the planning of computing resources for harvesting and normalising its metadata[2]. Also, it would be good to know in advance the number of records per resumption-token batch. This number can help us to choose the most suitable OAI *verb* for harvesting. For example, the verb "*ListIdentifiers*" may be preferable for a big batch size, rather than "*ListRecords*", or it may more efficient to harvest the repository by "*Sets*."

Another limitation of the OAI specification is the reality that a *Set* can be almost anything. A very basic subject-type standard for *sets* would at least make easier the identification of records that we really want to harvest. In many instances the "*Sets*" which are provided are more relevant to the internal organisational structure of the data provider rather than the potential needs of those harvesting the metadata. One more issue is keeping the federated searching databases containing harvested OAI metadata up-to-date. There is no way to be 100% sure of whether the OAI data provider is following the OAI "recommendations" for keeping its repository up-to-date, because they are just that - recommendations.

We have noticed that it is practically impossible to know for certain the type of information that many repositories are storing in their fields. The issue we are dealing with is the determination of the real content that the data providers have included in the tagged elements. Unqualified DC is a potential source for interpretation problems. The problem is that OAI data providers have total freedom to put anything in fields such as Author, Publisher, Abstract, Subject, Distributor, Identifier, etc.

These issues raise the need for a mechanism for assuring the completeness and quality of metadata harvested from OAI repositories. Completeness of metadata is a concept oriented to state which metadata elements are required for a particular type of resource to be usable for federated searching. For example, metadata about the author, journal ISSN and year are required for automated generation of openURL resolvers. Therefore in this case, metadata will be complete if all the elements required by openURL are included in the harvested metadata. Quality of metadata (Barton et al, 2003) is an even more challenging concept, since it is concerned with how to reconcile, enhance or correct metadata elements that are somewhat contradictory, incomplete or erroneous. Quality assurance and completeness agreements between OAI data providers and OAI services should be elaborated to advance research about the completeness and quality of OAI metadata.

2.3. SRU/SRW Protocol Issues

SRU and SRW have been developed mainly with the aim of simplifying some of the complexities involved with the Z39.50 protocol, while keeping the useful parts of the protocol, such as the CQL[3] query syntax. SRU/SRW services are easy to implement compared with Z39.50, mainly because SRU/SRW are Web-based protocols. Another important benefit is that SRU and SRW can be combined with other Web-driven applications such as OpenURL. Typically, SRU/SRW queries are encoded in URLs, the search results are in XML, and their records are encoded using the DC format.

---

[2] The Grainger Engineering Library Information Center at University of Illinois (USA) has created a OAI-PMH Data Provider Registry to alleviate the discovering task of OAI repositories [9]

[3] CQL: Common Query Language [10].

However, there are also some potential issues with these protocols. Thus the present version of the SRU protocol is bound to the HTTP GET operation, and is described in terms of a URI that includes query parameters. This use of HTTP GET subjects SRU to the same limitations suffered by HTTP GET (browsers limit the number of characters that can be included in a URI, and HTTP GET does not offer a character encoding mechanism of a URI, making it impossible for a CQL query to include non-ASCII characters). Recently, the Library of Congress has presented a specification to allow SRU requests to be expressed using HTTP POST. However it is not recommendable to support SRU/POST instead of SRU/GET, since the latter is in wide use. Also, a significant number of SRU/SRW implementations assume that anything received via POST is SRW while GET messages are assumed to be SRU. This heuristic can no longer be used. It becomes necessary to consult the Content-Type header, which is *text/xml* for SRW and *application/x-www-form-urlencoded* for SRU/POST. In conclusion, it seems that the simple SRU/GET is likely to continue to be favoured due its simplicity and suitability for services such as digital repositories. Federated searching developers need to keep in mind the limitations of these protocols.

3. Embedding federated searching toolkits in electronic learning environments

The availability of federated searching of external resources in virtual learning environments (VLEs) is essential in order to enable efficient information retrieval. However, in practice no VLE offers federated search facilities built into their native functionality. The absence of such functionality within VLEs restricts their ability for sharing and re-use of external e-learning sources discovered by federated searching. For example, there is a need for diverse digital repositories relevant to the user to be searched from within a VLE, and for the user to select and save search results within reading lists, which can then be made available to the rest of VLE components for learning activities. Unfortunately, such cross-domain use of digital repositories is not possible or it is too complex for the user to implement with current VLEs. We have studied the integration of federated searching in VLEs and its issues during the ELF Search Service Demonstrator Project [11], a project funded under the JISC e-Learning Framework Programme.

A low User-Interface level integration of federated searching with VLEs is not sufficient for re-using date discovered by federated searching. For example, both of the VLEs studied by the ELF Project, WebCT[12] and Moodle[13], offer the option to link a federated search service as an external URL or website framed within the VLE's framework. Moodle goes a little bit further and offers the option of integrating the federated searching into the underlying database as one of its modules or component. This option opens the possibility of saving and re-using results produced by federated searching for subsequent sharing with the rest of Moodle components. However, the setup of the mechanism to communicate with the Moodle back-end database is left to the users, which would be a difficult task for them to complete.

Various studies have shown growing consensus that an effective and feasible solution is to take a service-oriented approach for the development of e-learning infrastructures such as VLEs (Olivier 2005; Wilson, *et al*. 2004.) Technically this solution is within reach because of the increasingly widespread adoption of Web Services and Service-

Oriented Architectures (SOA). The Web Services standards are the best platform on which to build SOA infrastructures for VLEs. The monolithic and closed enterprise-level-applications architecture currently used by VLEs do not satisfy the learning and teaching needs of users. Using SOA, it is possible to eliminate the barriers that are precluding federated searching from being closely integrated within the native functionality of VLEs. A key point in this direction is the advocacy for the adoption of interoperability standards by the VLEs developers and vendors. Many efforts are underway to help make that happen. The e-Framework for Education and Research, which incorporates the E-Learning Framework (ELF), is an international effort to develop a service-orientated approach to the development and integration of applications in the sphere of e-learning. It has identified two levels of functional granularity: Learning Domains Services and Common Services. The former includes learning-specific components (assessment, course content management, resource lists, etc), while the latter identifies the underpinning cross-domain support services that are shareable among the learning domain services such as federated searching.

4. The PerX Pilot System

The PerX Project [14] has developed a pilot system to explore the practical issues that would be encountered when considering the possibility of full-scale subject resource discovery services. The pilot system integrates a federated searching toolkit, the PerX toolkit, which has been developed following a service-oriented model, and which is able to produce and consume XML. It can be deployed as a web services using the PHP *nuSOAP* library [15].

The Pilot provides a web interface [16] as a simplified and unified means for searching different type of repositories in engineering, such as:

- Scholarly bibliographic databases
- Institutional repositories
- Learning object repositories
- Industrial Technical Reports
- e-Journals articles
- Books

The PerX toolkit currently cross-searches OAI and Z39.50 repositories, the results from which are then aggregated for presentation to the user. Work is underway to support searching against SRU/SRW services. The toolkit technology uses the YAZ software [17] for searching Z39.50 databases and the search engine produced by the FAILTE Project [18] for searching OAI repositories.

4.1. Searching Z39.50 repositories with PerX

The PerX toolkit is able to cross-search the main Z39.50 software implementations, such as INNOPAC, Zebra, Endeavor, SIRSI, ALEPH and GEAC. A key aspect of the PerX toolkit is its ability to identify the "*attribute sets*" configuration for each Z39.50 target and to deal with them. Full information on the toolkit technicality can be found in the relevant web pages of the PerX project [19]. Basically our software uses a set of "query samples" to identify the configuration of the Z39.50 target. These "query samples" are built using the Prefix Query Format (PQF) structure, which abstracts the query string into an array of '*attributes*.' These attributes are collected together in '*attribute sets*.' The most commonly supported attribute set is Bib-1. We will present some examples of "samples queries" sent by the toolkit, using PQF, to uncover the *attribute sets* of Z39.50 database. For example, to search for the word *Java* in the title of records stored in the Z39.50 database, our '*sample query*' is:

@attrset bib-1 @attr 1=4 @attr 2=104 "Java" *(4.1.1)*

We also sample the prefixed Boolean operators (*@and*, *@or*, *@not*):

@attrset bib-1 @and @attr 1=4 @attr 2=104 "Java"
@attr 1=1003 @attr 2=104 "Morrinson" *(4.1.2)*

which will search for *Java* in the title AND *Morrinson* in the author fields. More advanced sample queries include the groupings of Boolean operators and combinations of attribute types, for example:

@attrset bib-1 @or @and @attr 1=4 @attr 2=3 @attr 3=1 @attr 6=1
"XML with Java" @attr 1=1016 @attr 2=104 @attr 3=3 @attr 6=1 "Morrinson"
@and @attr 1=62 @attr 2=104 @attr 3=3 "XML"
@attr 1=62 @attr 2=104 @attr 3=3 "Java" *(4.1.3)*

which will search for records with title equal to the phrase *"XML with Java*" AND "*Morrinson*" as author OR for records with the keywords *XML* AND *Java* in abstract.

The INNOPAC servers are among the most difficult to deal with, because they do not follow exactly the Z39.50 specifications. For example, the *Any Field* search (*use attribute* = 1016) is actually a *Title* search (*use attribute* = 4) on the INNOPAC Z39.50 server. Worse than that, the default configuration retrieves only titles in which the search term occurs at the beginning of the title. So the results of the Any Field search using the default INNOPAC setting will be extremely misleading for the user. Searching with truncation is also an issue. To know how a specific Z39.50 repository deals with truncation we need a trial and error process, which will eventually reveal whether or not search terms are being treated as if truncated. Some Z39.50 servers, including INNOPAC servers, accept the asterisk as a truncation symbol, which is not correct, because there is no provision in the Z39.50 standard for the asterisk to be used in this way.

INNOPAC Z39.50 server only supports Boolean searches with the *Any Field* search (*use attribute* = 4). But if the user tries to perform a Boolean search with other search types, the server simply converts those searches into Any Field searches, without warning the user of this, and it presents the corresponding results. This is misleading for the user. We have found that the following PQF *sample query* works better than the above (4.1.3) query with INNOPAC servers:

@or @and @attr 1=1003 @attr 1=21 @attr 1=4 @attr 3=3
"XML with Java" @attr @attr 1=1003 @attr 1=21 @attr 1=4
@attr 3=3 "Morrinson"
@and @attr 1=1003 @attr 1=21 @attr 1=4 @attr 3=3 "XML"
@attr @attr 1=1003 @attr 1=21 @attr 1=4 @attr 3=3 "Java" *(4.1.4)*

For some reason the use of these three different *Use Attributes* for each of the keywords makes a positive difference. A theorem proved on the basis of this work is that we never can assume that a Z39.50 server makes proper use of the standard *Bib-1* attributes. *Gils* is another commonly used attribute set supported by the PerX toolkit. There are more advanced specifications such as the *Attribute-architecture* set. However, while the *Attribute-architecture* is technically superior, only the most advanced Z39.50 implementations actively support it. None of the Z39.50 targets of interest to PerX are using it.

4.2. Harvesting, Transforming and Searching OAI repositories with PerX

Harvesting and transforming OAI repositories includes metadata harvesting, metadata normalization and metadata enhancement. These three processes use different software toolkits that run co-ordinately but separately.

Metadata harvesting. Full machine-to-machine harvesting is vital to keep an OAI service efficiently running and up-to-date. After trying with different Open Source OAI harvesters, the PerX Project took the decision to develop a software tool capable of harvesting any type of metadata. The native metadata schema can be Dublin Core (DC) or any richer metadata schema. Thus, the PerX OAI-PMH harvester is a flexible tool that can run with minimum human involvement and be adapted to deal with low volume or high volume of data. Its philosophy is to always try to complete the harvesting of a repository, regardless of whether the involved XML is valid or invalid, well-formed or ill-formed. The harvester generates logging information about the harvesting processes to help the maintenance and debugging tasks.

Metadata normalization. Metadata retrieved from OAI repositories conforms to a variety of implementations, which need to be made consistent and mapped to a common and unique XML structure that is used to render the search results on the user's browser. As has been noticed in Section 2.2, we cannot rely on the data harvested from OAI data providers. Without a normalization process, our federated searching service is exposed to all sorts of malfunctions. The OAI-PMH specification indirectly promotes the rapid and easy but unreliable production of OAI data provision by shifting to the OAI service providers the tasks of correcting and validating the metadata. We have met with a variety and significant number of errors being propagated from and by OAI data providers.

Metadata enhancement. The PerX OAI-PMH Harvester toolkit can enhance the metadata harvested from OAI repositories for metadata enrichment purposes in different ways. For example, it adds fields (e.g. the electronic type document), completes fields (e.g. full bibliographic reference), "cleans" fields (e.g. remove vCard tags), groups fields (e.g. description with notes) or splits fields (e.g. author's names into constituent parts for openURL construction), etc.

The PerX OAI-PMH toolkit is still under development. We are exploring full automatic metadata harvesting, normalization and enhancement. There are important cost advantages of automatic generation of searchable indexes over human controlled processes, which have been noticed by previous studies (Anderson and Perez-Carball, 2001; Greenberg et al, 2006). We are investigating "quick query identification" software toolkits, which can aid the metadata harvesting by providing an "at a glance" picture of the repository, with information about the supported metadata schemas, its sets, a profile of the content in the metadata fields, the expected number of records per resumption-token batch and, roughly, the number of records in the repository. We are also working on means for automatically spotting anomalies and characteristics in the harvested metadata. However, keeping a good channel of communication with the OAI data providers is advisable if that is possible. At the end of the day data providers know their data better than anyone.

5. Conclusions

Federated Searching by Subject has been implemented using the Z39.50 and OAI protocols and presented as a key function of VLEs, considering user studies that have shown its value (Richards, 2005.)

The sources and consequences of the inappropriate or conflictive implementations of the Z39.50 protocol have been studied. A primary consequence is that users receive imprecise search results or not find records even when they are in the databases. Other consequence is that implementers targeting Z39.50 products have to deal with issues that make difficult tasks such as finding how the Z39.50 attribute combinations have been implemented, and consequently how to construct effective search queries, or how to deal from a single-search point with databases that have implemented different indexes, etc. During the PerX project we have tried to overcome these obstacles by using in-house software toolkits. However we have reached the point where abandoning Z39.50 technology should be considered in favour of Web-based protocols.

It has been noticed that the OAI protocol is quite flexible in that there are relatively few mandatory specifications for implementation: valid responses to OAI verbs, the use of oai_dc, a unique and persistent OAI identifier, and a date-stamp. The rest of specifications are merely recommendations, and there is not a clear description of the consequences of not implementing some of the optional features of the protocol, which can be helpful for service providers. In addition, the need for establishing best practices for the metadata provided through OAI can be seen in the work that service providers must do to normalize and enhance their aggregations to ensure a certain threshold of usefulness for end-users. High quality ‘shareable’ metadata is crucial for a federated searching service.

The use of service-oriented architectures and web services has been suggested as a suitable open alternative to the closed environments of VLEs for embedding federated search toolkits. The core task of a VLE should be to provide a framework where service applications are embedded and integrated through agreed behaviours and interfaces using SOA technology to achieve systems integration.

It may be too late to prevent issues with the Z39.50 and OAI protocols from happening. We suggest the use of Web Services alternatives, such as SRU/SRW to satisfy the needs of the users of federated searching services. However it is not too late to reverse the tendency of VLE users, such as universities, committing themselves to enterprise level applications that do not support SOA or Web Services.

Acknowledgement

This work was supported by the PerX research project of the JISC-funded Digital Repositories Programme.

References

Stephen, T. and Harrison, T. (2002) “Building Systems Responsive to Intellectual Tradition and Scholarly Culture”. The Journal of Electronic Publishing.Vol 8 No. 1 http://www.press.umich.edu/jep/08-01/stephen.html

Hogg, M. and Field, J. "Using Z39.50 to build a virtual union catalogue Music Libraries Online: a subject clump.” Presented at the European Unicorn Users' Group conference, Sept. 2000, Madrid, Spain. Catalog-and-Index, no. 139, Spring 2001.

Lunau, Carrol D., “The Bath Profile: what is it and why should I care?” Library and Archives Canada. May 2003.
http://www.collectionscanada.ca/bath/ap-bathnew-e.htm

Lynch, Clifford A. “The Z39.50 Information Retrieval Standard. Part I: A Strategic View of Its Past, Present and Future”, D-Lib Magazine, April 1997.
http://www.dlib.org/dlib/april97/04lynch.html

Anderson, J.D. and Perez-Carball, J. (2001) ‘The nature of indexing: how humans and machines analyze messages and texts for retrieval – part I: research, and the nature of human indexing’, Information Processing and Management, Vol. 37, No. 2, pp.231–254.

Greenberg, J., Spurgin, K. and Crystal, A. (2006) ‘Functionalities for automatic metadata generation applications: a survey of metadata experts’ opinions’, Int. J. Metadata, Semantics and Ontologies, Vol. 1, No. 1, pp.3–20.
DOI: 10.1504/IJMSO.2006.008766

Olivier, B., T. Roberts, et al. (2005). “The e-Framework for Education and Research: An Overview”, JISC-CETIS (UK) DEST (Australia). www.e-framework.org

Wilson, S., Blinco, K., and Rehak, D. "Service-Oriented Frameworks: Modelling the infrastructure for the next generation of e- Learning Systems.” July 2004.
http://www.jisc.ac.uk/uploaded_documents/AltilabServiceOrientedFrameworks.pdf

Richards, G, Hatala, M. "Linking learning object repositories." International Journal of Learning Technology 2005 - Vol. 1, No.4 pp. 399 – 410
DOI: 10.1504/IJLT.2005.007151

Barton, J, Currier, S, and Hey, J. "Building Quality Assurance into Metadata Creation: an Analysis based on the Learning Object and e-Prints Communities of Practice", *DC-2003*
http://www.siderean.com/dc2003/201_paper60.pdf

Web-Links

[1] http://www.loc.gov/z3950/agency
[2] http://www.openarchives.org
[3] http://www.loc.gov/standards/sru
[4] http://www.icbl.hw.ac.uk/perx/analysis.htm

[5] http://www.indexdata.dk/yaz/doc/tools.tkl
[6] http://www.rdn.ac.uk
[7] http://www.iesr.ac.uk
[8] http://dublincore.org
[9] http://gita.grainger.uiuc.edu/registry
[10] http://www.loc.gov/standards/sru/cql
[11] http://www.icbl.hw.ac.uk/elfsearch
[12] http://www.webct.com
[13] http://moodle.org
[14] http://www.icbl.hw.ac.uk/perx
[15] http://dietrich.ganx4.com/nusoap
[16] http://www.engineering.ac.uk
[17] http://www.indexdata.dk/yaz
[18] http://www.failte.ac.uk
[19] http://www.icbl.hw.ac.uk/perx/tech/z39.50/queries